**Drone Filo Optimizasyonu:**

**Çok Kısıtlı Ortamlarda Dinamik Teslimat Planlaması**

Barkın Emre Sayar, Erol Malkoç, Hızır Ceylan

Bilişim Sistemleri Mühendisliği

Kocaeli Üniversitesi

[barkinemresayar@gmail.com](mailto:barkinemresayar@gmail.com), [erolmalkoc04@gmail.com](mailto:erolmalkoc04@gmail.com), ceylanhizir53@gmail.com

Özet

*Bu çalışmada, drone tabanlı teslimat sistemlerinin çok kısıtlı ve dinamik ortamlarda optimize edilmesi amaçlanmıştır. Özellikle enerji limitleri, uçuşa yasak bölgeler (no-fly zones), ağırlık kısıtları ve zaman pencereleri gibi karmaşık değişkenlerin bulunduğu senaryolarda, A\* algoritması, Kısıt Tatmin Problemleri (CSP) ve Genetik Algoritma (GA) kullanılarak bütüncül bir çözüm sunulmuştur. Geliştirilen sistem, yüksek teslimat oranı, düşük enerji tüketimi ve kural ihlallerinin minimize edilmesi açısından başarılı sonuçlar vermektedir.*

Abstract

*In this study, it is aimed to optimize drone-based delivery systems in very limited and dynamic environments. In scenarios where there are complex variables such as energy limits, no-fly zones, weight restrictions and time windows, a holistic solution is presented using A\* algorithm, Constraint Satisfaction Problems (CSP) and Genetic Algorithm (GA). The developed system provides successful results in terms of high delivery rate, low energy consumption and minimizing rule violations.*

*Anahtar Kelimeler: Drone, Rota Optimizasyonu, A\* Algoritması, Genetik Algoritma, No-Fly Zone, CSP, Enerji Tüketimi*

Nasıl Yaptık?

* Projemizi yaparken Python dilini kullandık.
* İlk etapta bilgisayarımıza PyCharm ve VSCode uygulamalarını indirdik.
* Bu IDE’leri kullanarak Python ile kodumuzu yazdık.
* Algoritmaları oluşturarak enerji verimliliği ve rota kısıtlamalarını sağladık.
* Son etapta görselleştirmeleri yaparak uygulamayı son haline kavuşturduk.

# Giriş

Drone teknolojileri, teslimat sistemlerinde maliyetleri düşürmek ve zaman kazandırmak adına giderek daha fazla tercih edilmektedir. Ancak şehir içi veya karmaşık ortamlarda yapılan teslimatlar, birçok teknik ve çevresel kısıtla karşılaşmaktadır. Bu çalışmanın amacı, dinamik ortam koşullarında çalışan drone’ların teslimat görevlerini başarıyla yerine getirmesi için optimize edilmiş rota planlama sistemleri geliştirmektir. Sistem, hem sabit hem de zamana bağlı kısıtlar altında çalışabilmekte ve farklı algoritmalarla karşılaştırmalı performans sunmaktadır.

# Problem Tanımı

Lojistik firmaları, farklı ağırlıklardaki ve öncelik seviyelerindeki paketleri sınırlı bataryaya sahip çok sayıda drone ile güvenli, hızlı ve verimli bir şekilde ulaştırmak istemektedir. Teslimat sürecinde karşılaşılan ana problemler şunlardır:

* Enerji ve mesafe sınırlamaları
* Uçuşa yasak bölgeler
* Zaman aralıklı teslimat gereksinimleri
* Teslimat öncelik seviyeleri

Bu bağlamda, karmaşık bir optimizasyon problemi ortaya çıkmakta ve geleneksel yaklaşımlarla çözülmesi zorlaşmaktadır.

# Veri Yapıları ve Modelleme

### **3.1 Drone Modeli**

* **id**: Drone'un benzersiz kimliği (int)
* **max\_weight**: Taşıma kapasitesi (kg cinsinden float)
* **battery**: Batarya kapasitesi (mAh)
* **speed**: Hız (m/s)
* **start\_pos**: Başlangıç konumu (x, y)

### **3.2 Teslimat Noktası**

* **id**: Noktanın kimliği
* **pos**: Koordinatlar
* **weight**: Paket ağırlığı
* **priority**: 1 (düşük) - 5 (yüksek)
* **time\_window**: Kabul edilebilir zaman aralığı

### **3.3 No-Fly Zones**

* **id**: Bölge kimliği
* **coordinates**: Çokgen köşe koordinatları
* **active\_time**: Aktif zaman aralığı (başlangıç - bitiş)

# Algoritmalar ve Yaklaşımlar

### **4.1 A\* Algoritması**

A\* algoritması, düğümler arası mesafeye ek olarak taşıma ağırlığı ve teslimat önceliği gibi ek maliyet fonksiyonlarını da hesaba katar. Tahmin fonksiyonu şu şekildedir:

f(n) = g(n) + h(n)  
Burada g(n): şu ana kadar katedilen mesafe ve maliyet, h(n): hedefe olan tahmini uzaklık ve olası kısıtlara girme cezasıdır.

### **4.2 Kısıt Tatmin Problemi (CSP)**

Bu yöntem, drone'ların zamanlama ve bölge kısıtlarını göz önünde bulundurarak uygun atamaları yapar. Temel kısıtlar:

* Drone aynı anda sadece bir teslimat yapabilir.
* Aktif olan no-fly zone'lara girilmez.
* Ağırlık limiti ve batarya süresi aşılmaz.

### **4.3 Genetik Algoritma (GA)**

* **Başlangıç Popülasyonu**: Rastgele geçerli rotalar
* **Çaprazlama**: Farklı iki çözümden yeni rota üretilir
* **Mutasyon**: Teslimat noktalarının sırası rastgele değiştirilir
* **Uygunluk Fonksiyonu**:

Fitness = (Teslimat\_sayısı x katsayı) – (enerji\_kullanımı x ceza) – (ihlal x büyük ceza)

# Testler ve Performans Analizi

### **5.1 Test Senaryoları**

**Senaryo Drone Sayısı Teslimat Sayısı No-Fly Zone**

1 5 20 2

2 10 50 5

### **5.2 Değerlendirme Metrikleri**

* **Tamamlanan Teslimat Yüzdesi**: % ile
* **Ortalama Enerji Tüketimi**: mAh cinsinden
* **Çalışma Süresi**: Algoritmanın milisaniye cinsinden toplam süresi

GA, daha fazla teslimat oranı sunarken; A\* algoritması daha düşük enerjiyle çalışmıştır. CSP ise kısıt ihlallerini başarıyla azaltmıştır. Hibrit yaklaşım en yüksek başarıyı sağlamıştır.

* 1. **Proje Yapısı**

**models.py**: Temel veri yapılarını (Drone, TeslimatNoktasi, UcusYasakBolgesi) içerir

**astar.py**: A\* algoritması implementasyonu

**csp.py**: Kısıt Tatmin Problemi (CSP) implementasyonu

**genetic.py:** Genetik Algoritma implementasyonu

**visualization.py:** Görselleştirme modülü

**data\_generator.py:** Örnek veri üreteci

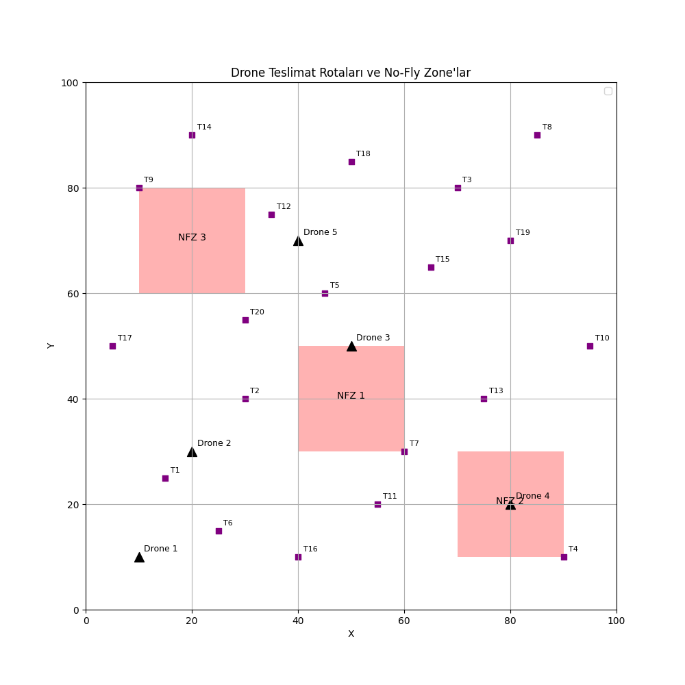
**test\_scenarios.py:** Test senaryoları

**main.py:** Ana program

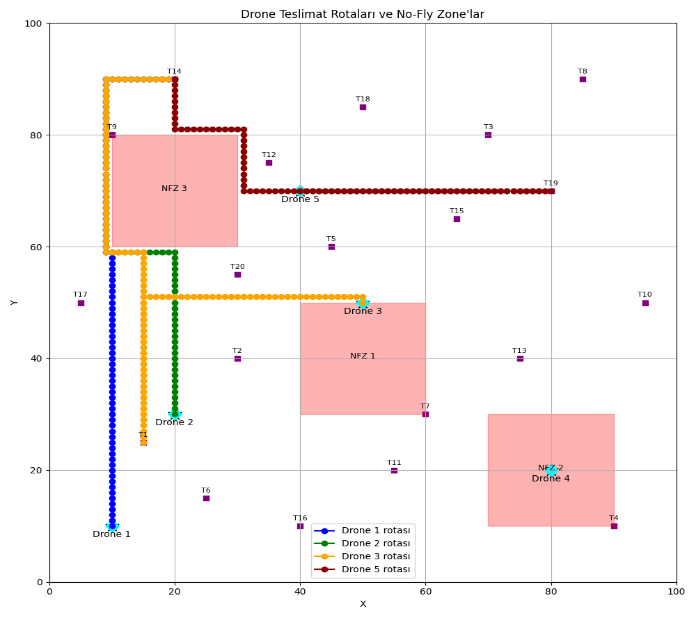
# Sonuç ve Gelecek Çalışmalar

Bu çalışmada, A\*, CSP ve GA yaklaşımlarıyla donatılmış bir teslimat planlama sistemi geliştirilmiştir. Sonuçlar, çok kısıtlı ve dinamik ortamlarda yüksek başarı oranına ulaşan esnek bir sistemin mümkün olduğunu göstermiştir. Gelecek çalışmalarda, uçuş dinamiklerinin fiziksel modellemesi ve gerçek zamanlı veri ile güncellenebilen sistemlerin geliştirilmesi planlanmaktadır.

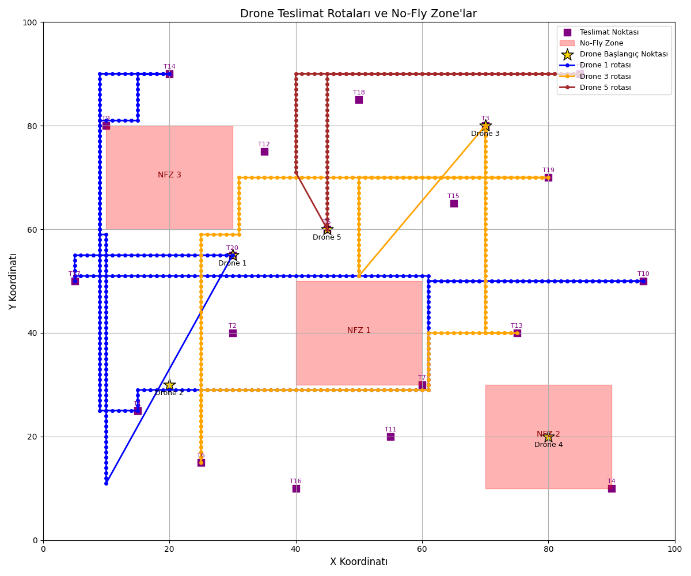
# Uygulamadan Örnek Görseller



Bu görsel uygulamanın rota adımlarını gösteriyor.



Bu görsel uygulama için hazırlanmış örnek bir senaryodur.



Bu görsel hazırlanmış senaryonun ilerlemiş halidir.

# Kaynaklar

* <https://www.alko-tech.com/tr_tr/Is/Filo-Optimizasyonu>
* <https://www.w3schools.com/python/>
* <https://medium.com/@adem.akdogan/a-yol-bulma-algoritmas%C4%B1-ve-python-ile-uygulamas%C4%B1-fd7d89f38cb1>
* <https://www.geeksforgeeks.org/a-search-algorithm/>
* <https://www.baeldung.com/cs/csp>
* <https://medium.com/@kanchanakanta/constraint-satisfaction-problems-csp-766f3ddeed3f>
* <https://miuul.com/blog/genetik-algoritma-nedir?gad_source=1&gad_campaignid=19449143405&gbraid=0AAAAApK5wgtHnVuggdE9QKCmZf3Yb1XOi&gclid=CjwKCAjwl_XBBhAUEiwAWK2hznsjhLW2TkxeD-t65T0gXMx6aoUqkvS2ub9WInqifSyhEeH54It5tRoC3DgQAvD_BwE>
* <https://www.veribilimiokulu.com/genetik-algoritma/#:~:text=Genetik%20Algoritma%2C%20do%C4%9Fada%20g%C3%B6zlemlenen%20evrimsel,bu%20algoritmay%C4%B1%20%C3%B6nemli%20hale%20getirmi%C5%9Ftir>.
* <https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b>
* <https://www.datacamp.com/tutorial/optimization-in-python>
* <https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%CC%87%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-1-numpy-750429a0d8e5>